2022

HTTP

UKA ZOTRIM

DTA |Infrastructure and Networks

History of HTTP

The development of the World Wide Web by Tim Berners-Lee and his group at CERN (Switzerland) in 1989 marked the beginning of HTTP. The name "One-line protocol" was given to the initial iteration of HTTP, which had the version number 0.9. This was limited to server-side HTML file viewing.

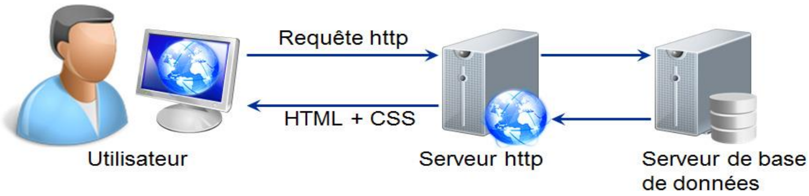
An application-level protocol for networked, collaborative, hypermedia information systems is the Hyper Text Transfer Protocol (HTTP).

HTTP 1.1 is the current industry standard. Then port 80 is used by an HTTP server by default (443 for HTTPS).

HTTP Protocol

A protocol called HTTP is used to retrieve resources like HTML documents. It serves as the foundation for all online data exchange. Since it is a client-server protocol, queries must be made by the recipient (which is usually a web browser). A complete document is constructed from several retrieved sub-documents, such as text, layout descriptions, photos, videos, scripts, and more.

In order to communicate, clients and servers exchange discrete messages (as opposed to a stream of data). Requests are messages delivered by the client, which is often a web browser, and replies are messages that the server sends back.



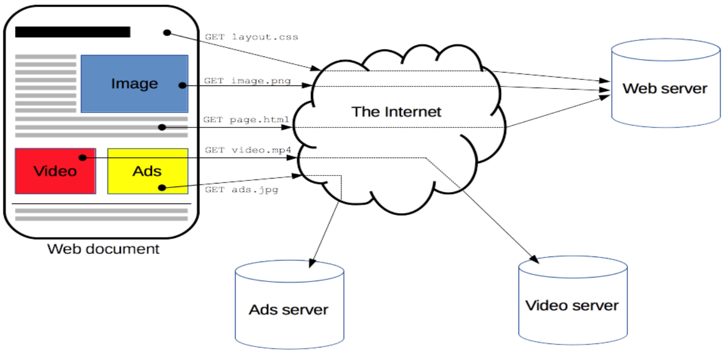
How it works

When you type an Internet address into your web browser and the requested website appears, HTTP connection between your browser and the web server has been established. We may therefore say that HTTP is the dialect that your web browser uses to communicate its requests to the web server.

1. The user enters information into his Internet browser's address bar.

2. The web server that manages the domain receives a request from the browser known as an HTTP request.

3. After receiving the HTTP request, the web server looks for the required file. and sends the header first. (The index.html file)



4. If the client wishes to obtain the file after it has been located. The server sends the message content following the header. (This is the file named index.html.)

5. After receiving the file, the browser presents it as a webpage.

Request method

The request methods defined by HTTP specify the action to be taken on the designated resource. Although there are nouns as well, we frequently discuss HTTP verbs. Differing implementations may have some common functionality despite having different meanings.

- Get : The GET method requests to represent the specified asset. GET requests should only be used to retrieve data.
- Head: The HEAD method requests a response identical to a GET request whose response body has been omitted (only the header).
- Post: The POST method is used to send an entity to the specified resource. This usually leads to a change of state or undesirable effects on the server.
- Put: The PUT method replaces the set of current representations of the resource covered by the content of the request.
- Delete: The DELETE method deletes the indicated resource.
- Trace : The TRACE method performs a round trip test message following the path of the targeted resource.

HTTP responses

A series of lines from the server known as an HTTP response are delivered to the browser. She is aware that

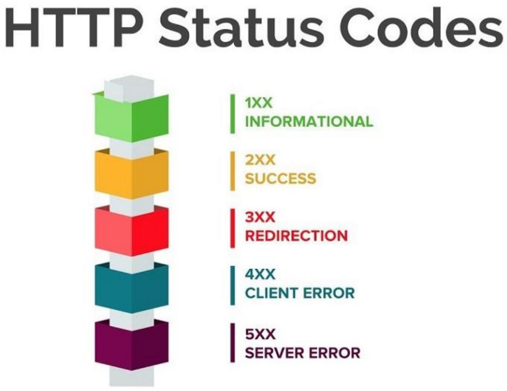
A status line is a line that indicates, via the use of a code and explanation text, the protocol version that is being used as well as the current stage of the request's processing. A gap is required between each of the three components on the line:

- The protocol's iteration that was employed
- The status code.
- What the code really means

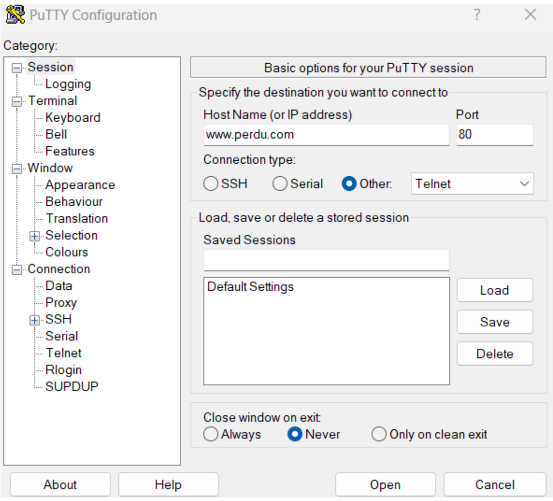
The response header fields are a collection of extra lines that can be included if further details about the response or the server are desired. These lines each consist of a name identifying the kind of header, a colon (:), and the header value. it contains the requested document.

there are 5 categories of http answers :

1. Informative answers (100 - 199)
2. Success answers (200 - 299)
3. Redirection messages (300 - 399)
4. Client errors (400 - 499)
5. Server errors (500 - 599)



An example of a GET request

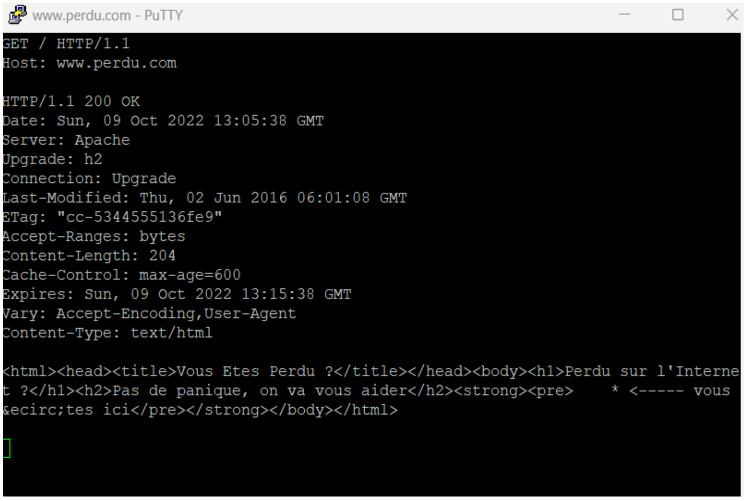


Host Name (or IP address): www.perdu.com

Port: 80

Connection type: ○ SSH ○ Serial ● Other: Telnet

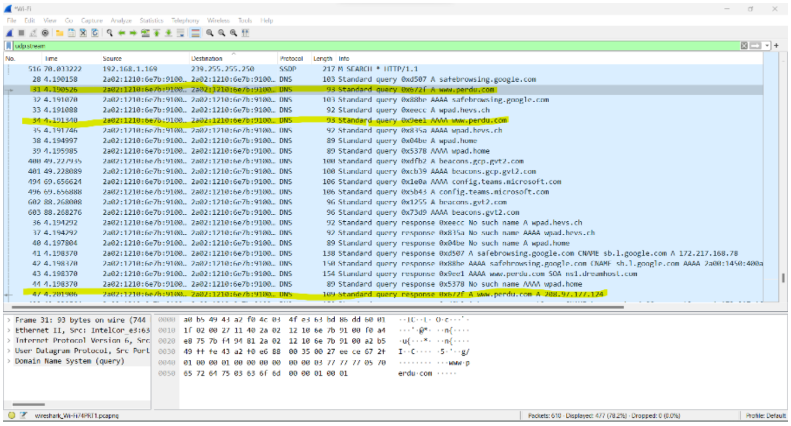
Close window on exit: ○ Always ● Never ○ Only on clean exit



```
GET / HTTP/1.1
Host: www.perdu.com

HTTP/1.1 200 OK
Date: Sun, 09 Oct 2022 13:05:38 GMT
Server: Apache
Upgrade: h2
Connection: Upgrade
Last-Modified: Thu, 02 Jun 2016 06:01:08 GMT
ETag: "cc-5344555136fe9"
Accept-Ranges: bytes
Content-Length: 204
Cache-Control: max-age=600
Expires: Sun, 09 Oct 2022 13:15:38 GMT
Vary: Accept-Encoding,User-Agent
Content-Type: text/html

<html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Interne
t ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>    * <----- vous
&ecirc;tes ici</pre></strong></body></html>
```


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

udp.stream

```
No.  Time       Source              Destination         Protocol Length Info
 516 70.033222  192.168.1.169       239.255.255.250     SSDP     217 M SEARCH * HTTP/1.1
  28 4.190158   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS  103 Standard query 0xd507 A safebrowsing.google.com
  31 4.190526   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS   93 Standard query 0x672f A www.perdu.com
  32 4.191070   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS  103 Standard query 0x88be AAAA safebrowsing.google.com
  33 4.191088   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS   92 Standard query 0xeecc A wpad.hevs.ch
  34 4.191340   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS   93 Standard query 0x9ee1 AAAA www.perdu.com
  35 4.191746   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS   92 Standard query 0x835a AAAA wpad.hevs.ch
  38 4.194997   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS   89 Standard query 0x04be A wpad.home
  39 4.195985   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS   89 Standard query 0x5378 AAAA wpad.home
 400 49.227935  2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS  100 Standard query 0xdfb2 A beacons.gcp.gvt2.com
 401 49.228089  2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS  100 Standard query 0xcb39 AAAA beacons.gcp.gvt2.com
 494 69.656624  2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS  106 Standard query 0x1e0a AAAA config.teams.microsoft.com
 496 69.656888  2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS  106 Standard query 0x5b43 A config.teams.microsoft.com
 602 88.268008  2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS   96 Standard query 0x1255 A beacons.gvt2.com
 603 88.268276  2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS   96 Standard query 0x73d9 AAAA beacons.gvt2.com
  36 4.194292   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS   92 Standard query response 0xeecc No such name A wpad.hevs.ch
  37 4.194292   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS   92 Standard query response 0x835a No such name AAAA wpad.hevs.ch
  40 4.197804   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS   89 Standard query response 0x04be No such name A wpad.home
  41 4.198370   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS  138 Standard query response 0xd507 A safebrowsing.google.com CNAME sb.l.google.com A 172.217.168.78
  42 4.198370   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS  150 Standard query response 0x88be AAAA safebrowsing.google.com CNAME sb.l.google.com AAAA 2a00:1450:400a
  43 4.198370   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS  154 Standard query response 0x9ee1 AAAA www.perdu.com SOA ns1.dreamhost.com
  44 4.198370   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS   89 Standard query response 0x5378 No such name AAAA wpad.home
  47 4.201906   2a02:1210:6e7b:9100.. 2a02:1210:6e7b:9100.. DNS  109 Standard query response 0x672f A www.perdu.com A 208.97.177.124
```

```
> Frame 31: 93 bytes on wire (744
> Ethernet II, Src: IntelCor_e3:63
> Internet Protocol Version 6, Src
> User Datagram Protocol, Src Port
> Domain Name System (query)
```

```
0000  a0 b5 49 43 a2 f0 4c 03  4f e3 63 bd 86 dd 60 01   ··IC··L· O·c···`·
0010  1f 02 00 27 11 40 2a 02  12 10 6e 7b 91 00 f0 a4   ···'·@*· ··n{····
0020  e8 75 7b f4 94 81 2a 02  12 10 6e 7b 91 00 a2 b5   ·u{···*· ··n{····
0030  49 ff fe 43 a2 f0 e6 88  00 35 00 27 ee ce 67 2f   I··C···· ·5·'··g/
0040  01 00 00 01 00 00 00 00  00 00 03 77 77 77 05 70   ········ ···www·p
0050  65 72 64 75 03 63 6f 6d  00 00 01 00 01            erdu·com ·····
```

Packets: 610 · Displayed: 477 (78.2%) · Dropped: 0 (0.0%)

Profile: Default

Sources

1. https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol
2. http://iamjmm.ovh/NSI/http/site/http.html
3. https://www.w3schools.com/tags/ref_httpmethods.asp
4. https://developer.mozilla.org/fr/docs/Web/HTTP/Overview
5. https://itnext.io/api-calls-and-http-status-codes-e0240f78f585
6. https://admaker.fr/blog/protocole-https-critere-referencement-pertinent/
7. https://www.1min30.com/dictionnaire-du-web/http
8. https://www.w3schools.com/whatis/whatis_http.asp